9. The machine-readable medium of claim 7 wherein said operations further comprise performing a memory alias analysis of said intermediate language program to partition the memory accesses of said intermediate language program into equivalence classes such that any two memory accesses that reference the same storage location belong to the same equivalence class.

10. The machine-readable medium of claim 8 wherein said operations further comprise performing a program analysis using said dependence flow graph.

11. The machine-readable medium of claim 10 wherein said program analysis comprises:

for each alias-node in the dependence flow graph assigning an initial value to the alias corresponding to said alias-node, and adding said alias-node to a set of nodes; and

while said set of nodes is not empty, iteratively performing the following:

removing a node from said set of nodes;

if said node is an alias-node then adding the successors of said node in the dependence flow graph to said set of nodes;

if said node is a definition-node for a statement of the form PUT (A, E) then determining a value for E, updating said initial value based on the value of E; and adding A to said set of nodes.

12. The machine-readable medium of claim 11 wherein said initial value comprises a set of abstract values which form a join-complete partial order.

13. An apparatus for doing program analysis comprising:

a processor and a memory coupled thereto having a set of instructions which when executed by the processor cause the processor to perform a method comprising:

assigning an alias to each equivalence class of possibly overlapping memory accesses as defined by an alias analysis of an intermediate language program; and

redefining a definition-use relationship between statements in each equivalence class wherein definition-statements which belong to the equivalence class reference the alias associated with that class and wherein use-statements which belong to said equivalence class reference the alias associated with that class.

14. The apparatus of claim 13 wherein said method further comprises constructing a dependence flow graph to represent said redefined definition-use relationship comprising:

assigning a definition-node for each definition statement in the program;

assigning a use-node for each use statement in the program;

assigning an alias-node for each alias;

introducing a single edge into the graph connecting each definition-node to its associated alias-node; and

introducing a single edge in the graph connecting each use-node to its associated alias-node.

15. The apparatus of claim 14 wherein said method further comprises first performing a memory alias analysis of said intermediate language program to partition the memory accesses of said intermediate language program into equivalence classes such that any two memory accesses that reference the same storage location belong to the same equivalence class.

16. The apparatus of claim 13 wherein said method further comprises performing a program analysis using said dependence flow graph.

17. The apparatus of claim 15 wherein said program analysis comprises:

for each alias-node in the dependence flow graph assigning an initial value to the alias corresponding to said alias-node, and adding said alias-node to a set of nodes; and

while said set of nodes is not empty, iteratively performing the following:

removing a node from said set of nodes;

if said node is an alias-node then adding the successors of said node in the dependence flow graph to said set of nodes;